Name: Vorname:

Geburtsdatum: Matrikelnummer:

Platz:

Nachhol-Klausur
Informatische Werkzeuge in den Geistes- und Sozialwissenschaften 1

21. April 2022

	Nur zur Korrektur, bitte freilassen!											
prob.	1.1	1.2	1.3	2.1	2.2	3.1	3.2	4.1	4.2	4.3	Summe	Note
total	4	4	6	4	4	6	6	4	12	10	60	
reached												

Klausurnote:

Bonuspunkte: Endnote:

Die „Lösungen“ der Aufgaben in diesem Dokument sollen den Studierenden als Anfangspunkt für die Beantwortung der Aufgaben dienen. Trotz aller Bemühungen kann es zu Unvollständigkeiten oder sogar Fehlern in den „Lösungen“ kommen. Da die Korrektur und Benotung der gestellten Aufgaben niemals lediglich auf einen „Vergleich mit der Musterlösung“ hinausläuft, ist dies auch nicht so schlimm. In jedem Fall sollten die Studierenden die Lösungen nachvollziehen und im Prinzip mittels des Lehrstoffs selbst verifizieren können.

Sollten Sie „Lösungen“ finden, die Sie nicht verstehen oder sogar für fehlerhaft halten diskutieren Sie diese am besten mit den Tutor*innen oder auf dem Kursforum und benachrichtigen Sie die Lehrenden; wir werden sie dann gegebenenfalls baldigst korrigieren.

1 Grundlagen und Verständnis

Aufgabe 1.1 (Dateipfade) 4 Pkt
Was ist ein *Dateipfad*? Wie unterscheidet sich dieser von einem *Dateinamen*? Was bedeuten in diesem Kontext *relativ* und *absolut*? 4 min

---

Solution: Ein Dateipfad ist eine Zeichenfolge die eine Datei oder ein Verzeichnis in einem Dateisystem beschreibt. Ein Dateiname ist teil eines Dateipfades, enthält aber keine Informationen darüber, in welchen Verzeichnissen (oder auf welchem Laufwerk) sich die Datei befindet.

Ein absoluter Pfad beschreibt den vollständigen Pfad vom Ursprung des Dateisystems und ist eindeutig. Viele Betriebssysteme erlauben allerdings auch, relative Dateipfade zu einem anderen Dateipfad (wie dem aktuellen Verzeichnis) zu benutzen, bei dem Teile des Pfades ausgelassen werden können und vom Betriebssystem aus dem Kontext ergänzt werden.

Korrektur:

- Je 1 Punkt für korrekte Erklärung von Dateipfad, Dateiname, relativ und absolut.

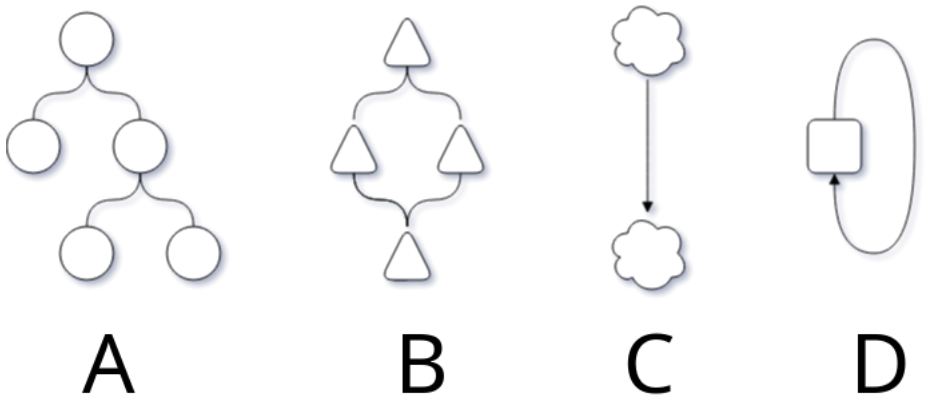
Aufgabe 1.2 (Server und Clients) 4 Pkt
Beschreiben Sie kurz die Rollen von „Servern“ und „Clients“ im World Wide Web und wie diese typischerweise miteinander interagieren und zu welchem Zweck. 4 min

---

Solution: Das World Wide Web basiert auf einer Server-Client Architektur in der Server Dateien und Dokumente (z.B. Videos oder Webseiten) zur Verfügung stellen und Clients (z.B. Browser) diese dem Nutzer oder der Nutzerin anzeigen. Normalerweise kommunizieren Server und Clients dabei über das HTTP (HyperText Transport Protocol).

Aufgabe 1.3 (Bäume) 6 Pkt
6 min

Welche der folgenden Graphen zeigen korrekte Bäume im mathematischen Sinne? Für nicht-Bäume begründen Sie bitte Ihre Entscheidung kurz (!).



Solution:

1. Baum
2. Kein Baum, Bäume erlauben es nicht, dass Elemente sich mehrere Vorfahren haben.
3. Baum
4. Kein Baum, Bäume erlauben es nicht, dass Elemente sich selbst zum Vor/-Nachfahren haben.

Korrektur: Je ein Punkt pro korrektem Baum/Nicht Baum. Bei nicht-Bäumen ein weiterer Punkt für eine korrekte Begründung.

2 Reguläre Ausdrücke

Aufgabe 2.1 (IBANs) 4 Pkt
4 min

In dieser Aufgabe geht es um einen regulären Ausdruck (Regex) für korrekte International Bank Account Numbers (IBANs). Die genaue Form von IBANs ist abhängig vom Land, wir suchen nach einem regulären Ausdruck für korrekte IBANS aus allen DACH-Ländern (d.h. Deutschland, Österreich und Schweiz).

Deutsche IBANs sind die Buchstaben "DE" gefolgt von 20 Ziffern, Österreichische IBANs sind die Buchstaben "AT" gefolgt von 18 Ziffern und Schweizer IBANs sind die Buchstaben "CH" gefolgt von 19 Ziffern.

Beispiele:

- DE89370400440532013000 (Deutschland)

- AT611904300234573201 (Österreich)
- CH9300762011623852957 (Schweiz)

Geben Sie einen regulären Ausdruck an, der genau alle DACH-IBANs erfasst.

Solution: Eine korrekte Lösung ist der reguläre Ausdruck:

`"DE\d{20}|AT\d{18}|CH\d{19}"`

Natürlich wäre auch ein Ausdruck wie `"DE\d\d\d\d\d\d\d\d\d\d\d\d\d\d\d\d\d\d\d\d..."` denkbar und technisch gesehen auch korrekt, aber solche Ausdrücken mangelt es an Klarheit für Dritte, die den Code lesen (inklusive Ihnen selbst in ein paar Wochen), sie sind anfälliger für Fehler und schwieriger zu debuggen.

Aufgabe 2.2 (Regulärer Ausdruck für Cryptocurrency Wallets) 4 Pkt

4 min

Als Teil einer Ermittlung gegen Geldwäsche werden Sie damit beauftragt, in einem großen Textkorpus alle potentiellen Wallet-Adressen für die Cryptowährung „PollutionCoin“ zu finden. So eine Adresse besteht für unsere Zwecke aus zwei Teilen (in dieser Reihenfolge), und zwar…

- …dem String `"1"`, `"3"` oder `"pc1"`
- …zwischen 27 und 34 (inklusive) Kleinbuchstaben, Großbuchstaben oder Ziffern. Dabei kommen die Buchstaben `I`, `l`, `O` und die Ziffer `0` nicht vor, um Verwechslungen auszuschließen.

Hier sehen Sie ein paar Beispiele für diese Adressen:

```
12cbQLTFMXRnSzktFkuoG3eHoMeFtpTu3S
1FeexV6bAHb8ybZjqQMjJrcCrHGW9sb6uF
3FfsbQntU2BAkVy2xaaRpSmLEDMEUwPbG1
pc1eHhgW6vquBYhwMPhQ668PjxTtpvZGPC
```

Geben Sie einen regulären Ausdruck an, der auf Wallet-Adressen wie oben beschrieben matcht. Falls Sie Gruppierungen verwenden, ist es egal ob diese *capturing* oder *non-capturing* sind.

Solution: Ein solcher Ausdruck wäre zum Beispiel:

```
([13]|pc1)[A-HJ-NP-Za-km-z1-9]{27,34}
```

Korrektur:

- Je einen Punkt für korrektes Matching des ersten Teils.
- Drei Punkte für korrektes Matching des zweiten Teils. Nur einen falls I, l, O und 0 nicht explizit ausgeschlossen.

---

3 Digitale Dokumente

Aufgabe 3.1 (Mentales CSS) 6 Pkt
Gegeben ist der Quelltext der folgenden **HTML**-Seite: 6 min

```
<!DOCTYPE html>
<html>

    <head>
        <style>
            p           {text-align: center;}
            body        {background-color: yellow;}
            h1          {font-size: 100pt;}
            span        {color: red;}
            div         {background-color: green;}
        </style>
    </head>

    <body>

        <h1 style="font-size: 5pt;">Headline</h1>

        <p>Lorem ipsum dolor sit amet...</p>

        <div>
            Lorem ipsum dolor sit amet...

            <form>
                <input type="submit" value="Useless button!">
            </form>

        </div>

    </body>
</html>
```

Beschreiben Sie, welche Elemente auf der fertig dargestellten **HTML**-Seite zu sehen sind, wie sie angeordnet sind und welche Farbe sie haben.

Solution: Die Website hat einen gelben Hintergrund und beginnt mit einer Überschrift (`<h1></h1>`). Die Überschrift lautet „Headline“ und ist nur in Schriftgröße 5 (nicht 100!).

Unter der Überschrift steht horizontal zentriert der Text „Lorem ipsum dolor sit amet...“.

Noch weiter unter befindet sich eine grüne Box (`<div></div>`). In dieser Box steht (nicht mehr horizontal zentriert) „Lorem ipsum dolor sit amet...“ .

Darunter, immer noch in der grünen Box ist ein klickbarer Knopf, der mit „Useless Button! “ beschriftet ist.

Kein Element hat die Klasse *span*, diese Regel findet also keine Anwendung.

Aufgabe 3.2 (XPath) 6 Pkt

6 min

1. Im Kontext von `XML`-Dokumenten, was versteht man unter einem *XPath*?
2. Beschreiben Sie außerdem für jeden der unten angegebenen XPaths, was genau damit erreicht werden kann.
   (a) `//*[not(*)]`
   (b) `//*[self::subject or self::note][2]/text()`

Solution: Die *XML Path Language* ist eine Abfragesprache für XML-Dokumente, ähnlich wie reguläre Ausdrücke für unstrukturierten Text. Ein XPath-Ausdruck adressiert eine bestimmte Menge an Knoten eines XML-Dokuments, das dabei als Baum betrachtet wird.

1. Dieser *XPath* beschreibt die Menge aller Knoten im Dokument, die selbst keine Nachfahren haben.
2. Dieser Knoten zielt auf nur den Text (nicht den ganzen Knoten!) von jeweils dem zweiten `subject` oder `note`-Knoten im ganzen Dokument, unabhängig von Vor- oder Nachfahren.

Korrektur:

- 2 Punkte für korrekte Beschreibung des Konzepts
- Je 2 Punkte für korrekte Beschreibung eines XPaths

---

4 Programmieren in Python

Aufgabe 4.1 (Python-Verständnisfrage) 4 Pkt

4 min

Gibt das folgende Programm etwas aus, wenn es ausgeführt wird? Oder wird die Schleife niemals gebrochen (und folglich nichts ausgegeben)? Begründen Sie Ihre Antwort.

```
einnahmen   = 1001
ausgaben    = 1000
angestellte = 1

quotient = einnahmen / ausgaben
profit   = 0

while True:

    for _ in range(angestellte):
        quotient = quotient * quotient

    einnahmen = einnahmen * quotient
    ausgaben  = ausgaben * (quotient / quotient)

    profit = einnahmen - ausgaben

    if profit > 0:
        angestellte += 1
    else:
        angestellte -= 1

    if profit > 1000000:
        break

print("Ziel erreicht!")
```

Solution: Es wird etwas ausgegeben, der `quotient` startet ≥ 1 und wächst von da an nur. Die Ausgaben bleiben konstant. Da der Gewinn wieder und wieder mit dem wachsenden Quotienten multipliziert wird, wächst er schnell auch über 1000000 (Die Schleife bricht im 5. Durchlauf).

Korrektur:

- Zwei Punkte für die Erkenntnis, dass es in der Tat Output gibt.
- Zwei weitere Punkte für korrekte Beschreibung, warum die Schleife nicht endlos ist.

Aufgabe 4.2 (Lotto) 12 Pkt

12 min

Beatrices Großvater hat mitbekommen, dass sie jetzt „Computer“ studiert und bittet sie um Hilfe beim Ausfüllen seines Lottoscheins. Leider lässt er sich nicht von ihren mathematischen Argumenten überzeugen, dass Lotterien wie diese statistisch identisch dazu sind, das Geld gleich zum Fenster hinaus zu werfen.

Basierend auf seinem Geburtsdatum (04.09.1947) und dem ihrer Großmutter (29.02.1948) möchte er die Zahlen `4,9,47,29,2,48` spielen. Er möchte allerdings wissen, ob diese Zahlen in genau dieser Kombination schon einmal gezogen wurden, und welche Superzahl am häufigsten gezogen wird.

Ihnen liegt eine CSV-Datei `lotto_archiv.csv` mit vergangenen Ergebnissen vor, die wie folgt formatiert ist:

```
KW;Datum;Gewinnzahlen;Zusatzzahl;Superzahl;Tag
1;01.01.2000;1,2,12,30,40,47;41;0;Samstag
2;08.01.2000;4,6,20,29,31,37;33;8;Samstag
3;15.01.2000;22,31,32,33,35,38;14;8;Samstag
4;22.01.2000;10,15,25,28,30,40;32;7;Samstag
5;29.01.2000;19,21,30,34,41,45;16;0;Samstag
[...]
```

Schreiben Sie ein python-Programm, das ausgibt, ob genau diese sechs Zahlen, die Beatrices Großvater spielen möchte, schon einmal so gezogen wurden. Wenn ja, dann soll auch das entsprechende Datum ausgegeben werden. Das Programm soll außerdem überprüfen (und ausgeben), welche Superzahl in der Datengrundlage am häufigsten vorkommt.

Hinweis:

- Die Reihenfolge der gezogenen Zahlen spielt bei dieser Lotterie für den Gewinn keine Rolle, für Ihre Abfrage allerdings schon.
- Sollten die Zahlen mehrmals vorgekommen sein oder mehrere Superzahlen gleich häufig erschienen sein, ist es egal welches Datum von den passenden Ziehungen / welche der häufigsten Superzahlen Sie ausgeben.
- Um einen String an einem Trennzeichen zu teilen, steht Ihnen die Funktion exampleString.split(trennzeichen) zur Verfügung.
  Beispiel: "3–1–4–1–5".split("–") == ["3", "1", "4", "1", "5"]

Solution: Hier ist eine mögliche Lösung:

```
# Geburtsdaten 29.02.1948 / 04.09.1947
# Wichtig: Aufsteigende Reihenfolge (!)
opasTipp = "2,4,9,29,48,47"

# Dictionary erstellen, am Anfang alles 0.
superzahlen = {}
for i in range(10):
    # Wir werden Strings einlesen, also muss dieses
    # Dictionary Strings als Schluessel haben.
    superzahlen[str(i)] = 0

# Wir gehen davon aus, dass Opas Tipp
# noch nie vorkam, bis wir es besser wissen.
datumVonTipp = None

with open("lotto_archiv.csv") as archive:
    # Erste Zeile brauchen wir nicht.
    headers = archive.readline()

    # Den Rest Zeile nach Zeile abarbeiten.
    for line in archive.readlines():
        splits    = line.split(";")
        numbers   = splits[2]
        superzahl = splits[4]

        # Check, ob diese Zahlen schon vorkamen.
        if numbers == opasTipp:
            datumVonTipp = splits[1]

        # Superzahl mit einbeziehen.
        superzahlen[superzahl] += 1

# Finden der frequentesten Superzahl
champ = "0"

for key in superzahlen:
    if superzahlen[key] > superzahlen[champ]:
        champ = key

# Ausgabe der Ergebnisse
if datumVonTipp is None:
    print("Opas Zahlen kamen noch nie vor!")
else:
    print("Opas Zahlen kamen zuletzt am", datumVonTipp, "vor!")

print("Die meistgezogene Superzahl war", champ)
```


Korrektur

- Jeweils 6 Punkte für die korrekte Ermittlung des Datums und der häufigsten Superzahl.
- 2 Punkte Abzug, falls nicht beachtet wird, dass die Zahlen unsortiert gegeben aber sortiert verglichen werden.

Aufgabe 4.3 (Python-Programmieraufgabe bottle / Liste) 10 Pkt

Geben Sie den python-Quellcode für eine `bottle`-Route an, die auf dem Pfad 10 min

/echo/...

ansteuerbar ist und zwei Parameter entgegen nimmt..

Die Route soll eine valide HTML-Seite zurückgeben, deren `<body>`-Tag lediglich eine nicht nummerierte Liste enthält. Diese Liste soll so viele Elemente haben, wie der zweite Parameter angibt. Diese Elemente sollen jeweils den ersten Parameter als Text enthalten.

Dabei soll das erste Element eine Schriftgröße haben, die dem zweiten Parameter gleicht, das zweite eine Schriftgröße kleiner und so weiter bis runter auf 1.

Beispiel: /echo/Hallo/15/ würde also eine HTML-Seite mit einer Liste zeigen, die 15 Elemente mit dem Text „Hallo“ hat, in den Schriftgrößen 15, 14, 13, ..., 3, 2 und 1.

Hinweis: Zur Erinnerung: Sie können CSS-Regeln für einzelne HTML Knoten wie folgt über das `style`-Attribut des Knotens setzen:

```
<div style="color:red;">Here's some red text in a div!</div>
```

Solution: Eine Lösung ist diese hier:

```
from bottle import route, run

@route('/echo/<text>/<size>/')
def colour(text, size):

    elements = ""

    # Prepare list elements
    for i in range(int(size)):
        cursize = str(int(size) - i) + 'pt'
        elem = '<li style="font-size:' + cursize + '">' + text + '</li>\n'
        elements = elements + elem

    # Compose HTML
    html = """<!DOCTYPE HTML>
    <html>
        <body>
            <ul>""" + elements + """</ul>
        </body>
    </html>
    """

    # Return HTML
    return html

run(host='0.0.0.0', port=32500)
```

Korrektur:

- 2 Punkte für korrekte Definition von Route und Funktion
- 2 Punkte für eine korrekte Liste mit richtig vielen Einträgen
- 4 Punkte für die korrekte Benutzung und Verringerung der Schriftgröße
- 2 Punkte für Rückgabe von HTML in dem die Liste korrekt eingebunden wurde.